

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 9月11日

出 願 番 号

Application Number:

特願2002-264977

[ST.10/C]:

[JP2002-264977]

出 願 人

Applicant(s):

株式会社東芝

2003年 2月14日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎

出証番号 出証特2003-3007463

【書類名】 特許願
 【整理番号】 13B0280691
 【あて先】 特許庁長官殿
 【国際特許分類】 G09C 1/00
 【発明の名称】 暗号演算回路
 【請求項の数】 5

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

【氏名】 藤崎 浩一

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

【氏名】 新保 淳

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

【氏名】 本山 雅彦

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

【氏名】 池田 華恵

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
 研究開発センター内

【氏名】 友枝 裕樹

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100083161

【弁理士】

【氏名又は名称】 外川 英明

【電話番号】 (03)3457-2512

【手数料の表示】

【予納台帳番号】 010261

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 暗号演算回路

【特許請求の範囲】

【請求項 1】 所定ビット数の第 1 データブロックと鍵情報とを所定の非線形関数 F へ作用させた結果得られる結果値と該所定ビット数の第 2 データブロックとを論理演算して結果値を得て、前記第 2 データブロックを第 1 データブロックと置き換えるとともに、前記第 1 データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、

前記非線形関数 F を作用させるための前記鍵情報を前記暗号化処理とは無関係の情報として供給する手段を備え、前記暗号化データの出力の後に、前記供給する手段によって開始するようにしたことを特徴とする暗号演算回路。

【請求項 2】 所定ビット数の第 1 データブロックと鍵情報とを所定の非線形関数 F へ作用させた結果得られる結果値と該所定ビット数のデータブロックとを論理演算して結果値を得て、前記第 2 データブロックを第 1 データブロックと置き換えるとともに、前記第 1 データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、

前記第 1 データブロックを保持する保持手段と、

この保持手段へ前記暗号化処理とは無関係の情報を供給する手段を備え、前記暗号化データの出力の後には、前記供給する手段によって供給を開始するようにしたことを特徴とする暗号演算回路。

【請求項 3】 所定ビット数の第 1 データブロックと鍵情報とを所定の非線形関数 F へ作用させた結果得られる結果値と該所定ビット数の第 2 データブロックとを論理演算して結果値を得て、前記第 2 データブロックを第 1 データブロックと置き換えるとともに、前記第 1 データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、

前記非線形関数 F を作用させるための前記鍵情報を前記暗号化処理とは無関

係の情報として供給する第 1 供給手段と、

前記第 1 データブロックを保持する保持手段と、

この保持手段へ前記暗号化処理とは無関係の情報を供給する第 2 供給手段とを
備え、

前記暗号化データの出力の後には、前記第 1 供給手段および前記第 2 供給手段
によってそれぞれ供給を開始するようにしたことを特徴とする暗号演算回路。

【請求項 4】 非線形関数 F を処理する手段を備える F e i s t e l 型暗号アル
ゴリズムを実装し、暗号化データを出力する暗号演算回路であって、

前記非線形関数 F を処理する手段へ暗号演算結果とは無関係のデータを供給す
る手段を備え、前記暗号化データの出力の後に、前記供給する手段によってデー
タ供給を開始するようにしたことを特徴とする暗号演算回路。

【請求項 5】 F e i s t e l 型暗号アルゴリズムを実装し、暗号化データを出力
する暗号演算回路であって、

暗号演算回路内の内部データを保持する保持手段と、

この保持手段へ前記暗号化処理とは無関係の情報を供給する手段とを備え、

前記暗号化データの出力の後には、前記供給する手段によって供給を開始する
ようにしたことを特徴とする暗号演算回路。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、F e i s t e l 型アルゴリズムを実装した暗号演算回路に関し、特
に、電力解析攻撃に対する耐性を持った暗号化演算回路に関する。

【 0 0 0 2 】

【従来の技術】

暗号演算回路に対して演算処理中の消費電力に関するグラフを作成し、その消
費電力グラフを用いて、暗号演算回路に納められている鍵情報を取り出すという
電力解析攻撃(D P A: Differential Power Analysis) 法がある。

【 0 0 0 3 】

D P A 法は、まず、複数のデータをそれぞれ暗号演算回路に入力し、それぞれ

の消費電力を測定する。次に、暗号演算回路内にある鍵情報の推定を行う。このとき、推定した鍵情報と消費電力グラフに相関がある場合、計算されたその相関値は、推定した鍵情報が正しい時には大きな値となり、推定した鍵情報が間違っている時には小さな値となる。DPA法は、この原理を用いて、消費電力を測定することにより暗号演算回路内部の鍵情報を取り出すという攻撃方法である。

【 0 0 0 4 】

DPA法は、破壊行為を伴わない攻撃であるため、外見をみただけでは攻撃によって鍵情報が取り出されたかどうか判断できず、不正利用による被害が拡大する恐れがある。このため、暗号演算回路において、DPA法への対策が必須となってきた。

【 0 0 0 5 】

DPA法に対する対策の一つとして、演算中のデータをマスクして演算する発明があった（例えば、特許文献1参照）。この発明は、鍵情報を用いた演算中の消費電力の変動が鍵情報と関係しないように、鍵情報をマスクして演算を行なうというものである。この発明は、鍵情報をマスクして演算することで、鍵情報と消費電力との相関関係を失わせるようにして、DPA法対策を行ったものである。

【 0 0 0 6 】

【特許文献1】特開2000-66585公報

【 0 0 0 7 】

【発明が解決しようとする課題】

ところで、共通鍵ブロック暗号方式では、Feistel氏の開発したFeistel型暗号アルゴリズムを用いたものが多く採用されている。

【 0 0 0 8 】

従来の技術欄で説明した、特開2000-66585公報の発明は、このFeistel型暗号アルゴリズムを実装した暗号演算回路において、演算終了時点のDPA法の対策として有効であるが、演算終了の後の回路動作に対して行われるDPA法に対する対策では無かった。

【 0 0 0 9 】

そこで、本発明は、このような問題点に鑑みなされたものであり、Feistel型暗号アルゴリズムを実装した暗号演算回路の演算終了の後に行われるDPA法の対策を実現した暗号演算回路を提供することを目的とする。

【0010】

【課題を解決するための手段】

上記目的を達成するために、本発明は、所定ビット数の第1データブロックと鍵情報とを所定の非線形関数Fへ作用させた結果得られる結果値と該所定ビット数の第2データブロックとを論理演算して結果値を得て、前記第2データブロックを第1データブロックと置き換えるとともに、前記第1データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、前記非線形関数Fを作用させるための前記鍵情報を前記暗号化処理とは無関係の情報として供給する手段を備え、前記暗号化データの出力の後に、前記供給する手段によって始するようにした。

【0011】

また、本発明は、所定ビット数の第1データブロックと鍵情報とを所定の非線形関数Fへ作用させた結果得られる結果値と該所定ビット数のデータブロックとを論理演算して結果値を得て、前記第2データブロックを第1データブロックと置き換えるととともに、前記第1データブロックを前記結果値と置き換える攪拌ステップを繰り返し実行することにより、暗号化処理を行って、該処理結果を暗号化データとして外部へ出力する暗号演算回路であって、前記第1データブロックを保持する保持手段と、この保持手段へ前記暗号化処理とは無関係の情報を供給する手段を備え、前記暗号化データの出力の後には、前記供給する手段によって供給を開始するようにした。

【0012】

本発明は、例えば、Feistel型暗号アルゴリズムを実装した暗号演算回路の信号線の遷移や信号線の値に着目し、それぞれにおいてDPA法により攻撃されるタイミングと攻撃ポイントを明確にして、上記のように対策するようにしたから、DPA法による攻撃に対して暗号演算回路の内部にある鍵情報の漏洩を防ぐ

ことができる。

【 0 0 1 3 】

【発明の実施の形態】

以下、本発明の実施の形態について、図面を用いて詳細に説明する。

【 0 0 1 4 】

まず、Feistel型暗号アルゴリズムについて、その処理の流れを図1のフローチャートを用いて簡単に説明する。

【 0 0 1 5 】

まず、暗号化する入力ブロックデータは、初期転値(I P:Initial Permutation)と呼ばれる、入力ブロックデータのビット間のデータを並び換える処理が施される(ステップS1)。次に、この初期転値後のブロックデータを $n/2$ ビットのブロックに分割する(ステップS2)。この分割された2つのブロックを初期値とし、以下の演算を繰り返す(ステップS3、S4)。

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \wedge F(\text{key}_i, R_{i-1})$$

ここで、演算回数を i ($1 \leq i \leq k$)、 i 回目の演算時の一方の分割されたブロック(右側)を R_i 、他方のブロック(左側)を L_i 、 i 回目の鍵情報を key_i 、非線形演算を行なう関数を F とする。また、「 \wedge 」は、排他的論理和を示す。

【 0 0 1 6 】

この繰り返し演算S3を k 回繰り返した後(S4のY)に、各演算結果 R_k 、 L_k を合成する(ステップS5)。なお、この合成は、 L_k を上位 $n/2$ ビットのブロックに、 R_k を下位 $n/2$ ビットのブロックとした n ビットのブロックデータとすることを意味する。そして、合成されたブロックデータに対し、逆初期転値(IP^{-1})を行ない(ステップS6)、演算結果を得て演算を終了する。

【 0 0 1 7 】

次に、このFeistel型暗号アルゴリズムを実装した暗号演算回路のデータパス部の構成を図2に示す。

【 0 0 1 8 】

IP部21は、予め定められた規則に基づいて、初期転値の演算を行なう回路

モジュールである。関数演算部 2 4 は、所定の非線形関数である関数 F の演算を行なう回路モジュールである。 IP^{-1} 部 2 5 は、 IP 部 2 1 で定められた規則に相反する規則に基づいて、逆初期転値の演算を行なう回路モジュールである。 L レジスタ 2 3、 R レジスタ 2 2 は、それぞれ $n/2$ ビット幅を持ったレジスタであり、 3 入力 1 出力のセクタ 2 6、 2 7 は、図示されていない制御回路部からの信号に応じて、 L レジスタ 2 3、 R レジスタ 2 2 へ入力するデータを選択する。なお、セクタ 2 6 への 3 入力とは、 IP 部 2 1 からの上位 $n/2$ ビットの値と、 R レジスタ 2 2 の出力値と、 L レジスタ 2 3 の出力値と関数演算部の出力値とを排他的論理和した値とであり、セクタ 2 7 への 3 入力とは、 IP 部 2 1 からの下位 $n/2$ ビットの値と、 R レジスタ 2 2 の出力値と、 L レジスタ 2 3 の出力値と関数演算部の出力値とを排他的論理和した値とである。

【 0 0 1 9 】

さて、以上説明した構成において処理された暗号演算結果 $C = (C_l, C_r)$ は、次式のように表すことができる。

$$C = IP^{-1} \{ (L_k \wedge F(key_k, R_k)), R_k \}$$

ここで、 IP^{-1} 部 2 5 は、ビット並びを入れ換えているだけであるために、演算結果 C から R_k の値は容易に特定される。 R_k の値が特定されると、演算終了後の関数演算部 2 4 の出力値は、鍵情報 key_k の一変数の値のみによって決まる。以上のことから、関数演算部 2 4 の出力結果を攻撃ポイントとして、鍵情報 key_k を推定することにより、 DPA 法が行われてしまう。仮に、関数演算部 2 4 の消費電力の変動と鍵情報の演算との間に相関があった場合には、攻撃者に鍵情報を特定されてしまう。すなわち、暗号演算の終了時点で、 DPA 法を用いた攻撃を行なうことで、暗号演算回路内の鍵情報が取り出されてしまう。この DPA 法を用いた攻撃法の対処方法としては、先に示した特開 2 0 0 0 - 6 6 5 8 5 号公報による発明によって解決できる。

【 0 0 2 0 】

一方、Feistel型暗号アルゴリズムを回路実装した暗号演算回路においては、上記の暗号演算の終了時点での攻撃の他に、暗号演算が終了後（終了時点の次のタイミング以降）においても、 DPA 法を用いた攻撃を行うことで、暗号演算回

路内の鍵情報が取り出されてしまう可能性があることが新たに分かった。このことを説明する前に、暗号演算回路の演算結果を出力する際のデータパス部におけるデータの制御方法には2通り有って、各制御方法によって、暗号演算の終了後のDPA法による攻撃ポイントが異なっているため、まず、2通りのデータの制御方法について説明する。

【0021】

1つめの制御方法は、演算結果を出力するタイミングにおいて、図3にあるようにセレクタ27がデータパス28を選択し、かつセレクタ26がデータパス29を選択することで、左右のデータを入れ換えた形で演算を終わらせる、クロス型回路構成である。一方、2つ目の制御方法は、演算結果を出力するタイミングにおいて、図10にあるようにセレクタ27がデータパス29を選択し、セレクタ26がデータパス28を選択することで、演算の最後において左右のデータを入れ換ええないような形で演算を終わらせる、ストレート型回路構成である。

【0022】

Feistel型暗号アルゴリズムを実装した暗号演算回路において、そのアルゴリズムで定められている繰り返し演算を行なう場合、セレクタ26、セレクタ27は左右のデータが入れ替わるように、図2において図示されていない制御回路により制御される。この暗号演算回路において、演算結果を出力する場合に、特に必要がない場合は繰り返し演算を行なう場合と同様に、左右のデータが入れ替わったクロス型回路構成で演算を終了させる。このようなクロス型回路構成とする理由として、演算結果を出力するタイミングで演算結果を出力するタイミングでデータの流れを変えるように制御するよりも、演算結果を出力する場合であっても、繰り返し演算を行なっているときと同様に左右のデータが入れ替わるような制御を行なった方が、制御が簡単であり制御回路の実装も小さくできるからである。

【0023】

一方、DESを3回行なうTriple DES(3DES)を実装した回路の場合には、1回目のDESの演算結果と2回目のDESの演算開始間、および2回目のDESの演算結果と3回目のDESの演算開始間において、左右のデータを入れ換ええない

ような制御を行なう必要がある。そのため、3 回目の DES が終了後も左右のデータを入れ換えずに出力するストレート型回路構成とすることが多い。このような回路構成となるのは、暗号演算回路として回路実装する場合において、その回路規模を小さくするために、演算中にできる限り場合分けを減らすような制御を行なうようにするためである。1 回目、2 回目が終わったときと同様に、3 回目が終わったときにもストレート型回路構成となるようにセレクタ 26、セレクタ 27 を制御したほうが、場合分けの数が少なくなり制御回路を小さく構成することができる。

【 0 0 2 4 】

以上説明したように、Feistel 型暗号アルゴリズムを実装した場合には、クロス型回路構成とストレート型回路構成との何れかで構成される。

【 0 0 2 5 】

次に、これら回路構成の違いによる演算終了後における DPA 法による攻撃ポイントについて説明する。

【 0 0 2 6 】

まず、クロス型回路構成の場合について、DPA 法により攻撃のポイントとなるタイミングと、その原因について説明する。

【 0 0 2 7 】

演算結果を出力するクロック Clock_{fin} の立ち上がりのタイミングにおいて、L レジスタ 23 の値を L_{fin} 、R レジスタ 22 の値を R_{fin} とする。このとき、この暗号演算回路の演算結果を C とすると、 C は C_l と C_r を用いて $C = IP^{-1}(L_{fin} \wedge F(\text{key}_{fin}, R_{fin}), R_{fin})$ と表すことができる。ここで、 IP^{-1} は、ビットの並びを変換する処理だけであるため、演算結果 C から R_{fin} と $L_{fin} \wedge F(\text{key}_{fin}, R_{fin})$ の値は簡単に解ってしまう。

【 0 0 2 8 】

次に、クロック Clock_{fin+1} において、R レジスタ 22 に値が書き込み可能である場合、R レジスタ 22 の値は $L_{fin} \wedge F(\text{key}_{fin}, R_{fin}) = R_{fin+1}$ となる。このとき、鍵情報が Clock_{fin} と同じ key_{fin} であるので、関数演算部 24 の出力は $F(\text{key}_{fin}, R_{fin+1})$ となる。 R_{fin+1} は、暗号演算回路の出力結果から簡単に解つ

てしまう値であり、そのため、関数演算部 2 4 の関数 F は、 key_{fin} の一変数の関数と見なすことができる。

【 0 0 2 9 】

以上のように、関数演算部 2 4 の出力の遷移を攻撃ポイントとして、関数演算部 2 4 の出力を推定することにより鍵情報を特定されてしまうといった問題点があった。

【 0 0 3 0 】

また、上記の他に、L レジスタ 2 3 の値の遷移を攻撃ポイントとされる別の問題点がある。これについて説明する。

【 0 0 3 1 】

$Clock_{fin}$ 、 $Clock_{fin+1}$ における L レジスタ 2 3 の値を L_{fin} 、 L_{fin+1} とする。
L レジスタ 2 3 の値の遷移は次式で表すことができる。

$$L_{fin} \wedge L_{fin+1} = R_{fin-1} \wedge R_{fin}$$

なお、上記式の右辺は、クロス型回路構成において L レジスタ 2 3 の値は $L_i = R_{i-1}$ ($1 \leq i \leq k$) という関係があることから、書き変えたものである。ここで、 R_{fin} は外部から観測可能な値であることから、 R_{fin-1} に対して推定することができる。L レジスタ 2 3 の遷移は、暗号演算回路の消費電力の変動と関係があるので、DPA 法により R_{fin-1} の値が特定される可能性がある。つまり、上記で述べたように演算結果 C から、 $L_{fin} \wedge F(key_{fin}, R_{fin}) (= R_{fin+1})$ および R_{fin} の値は観測可能である。ここで、 $L_{fin} = R_{fin-1}$ であることから、 R_{fin-1} が特定されると、 R_{fin+1} 、 L_{fin} 、 R_{fin} が知ることのできる値となるため、 key_{fin} を特定されてしまうという問題点があった。

【 0 0 3 2 】

以上のように、関数演算部 2 4 の出力値の遷移、または、L レジスタ 2 3 の値の遷移において、DPA 法を用いて攻撃される可能性がある。

【 0 0 3 3 】

クロス型回路構成の場合のこれら DPA 法による攻撃に対する対策は、それぞれ異なる。

【 0 0 3 4 】

まず、このクロス型回路構成の Clock_{fin+1} のタイミングにおける関数演算部 24 の出力値に関する DPA 対策として、本実施の形態においては、 Clock_{fin+1} のときに関数演算部 24 に入力される値は、鍵情報と関係のない値を入力するような回路構成とする。このような回路の具体例としては、図 4 に示すように関数演算部 24 へ鍵情報を入力する個所へ新たにセレクタを設け、 Clock_{fin} までは鍵情報を入力し、 Clock_{fin+1} 以降は乱数を入力するような回路構成や、図 5 にあるように関数演算部 24 の入力として、鍵情報と乱数を排他的論理和 (XOR) 演算、論理積 (AND) 演算、または論理和 (OR) 演算した値を入力する回路構成とし、 Clock_{fin+1} 以降のクロックの立ち上がり時に、常に異なる乱数を与えて得た演算結果を入力するという回路構成で解決した。

【 0 0 3 5 】

また、関数演算部 24 の入力として、外部から観測可能な値を入れないようにする回路構成も対策となりうる。このような対策の具体例としては、図 6 に示すように、関数演算部 24 の入力の前にセレクタを設け、 Clock_{fin+1} 以降は乱数を入力するような回路構成を用いる例がある。また、図 6 のセレクタを用いて乱数を入れる代わりに、図 7 のように演算回路を追加し、R レジスタ 22 の値と乱数との演算結果が、関数演算部 24 に入力されるような回路構成としてもよい。このように Clock_{fin+1} 以降、関数演算部 24 の入力として、暗号演算回路で利用した鍵情報や演算結果が、そのまま入力されないような回路構成とすることで、関数演算部 24 の出力における DPA 対策とすることができる。

【 0 0 3 6 】

このほかに、 Clock_{fin+1} 以降、R レジスタ 22 の値を、演算結果と無関係な値とすることで、DPA 法対策ができる。例えば、図 8 にあるように R レジスタ 22 の入力の前にセレクタを入れ、 Clock_{fin+1} 以降、このセレクタから乱数が R レジスタ 22 に書き込まれるような回路構成や、図 6 のようにデータパス 29 の結果と乱数を演算するような回路構成とすればよい。

【 0 0 3 7 】

以上のような対策を施した暗号演算回路とすることにより、関数演算部 24 の出力の遷移を攻撃のポイントとできなくなった。

【 0 0 3 8 】

次に、Lレジスタ23の値の遷移を攻撃ポイントとした場合には、 $Clock_{fin+1}$ のタイミングでLレジスタ23に R_{fin+1} を書き込まないような回路構成としなければならない。この時の対策として、回路構成の具体例は、図11のようにLレジスタ23へのデータパス30上にAND/OR/XORなどの演算回路を入れ、 $Clock_{fin+1}$ 以降でデータパス30上の値と乱数の演算結果をLレジスタ23に書き込むという方法や、図12にあるようにデータパス30にセクタを入れることで $Clock_{fin+1}$ 以降、Lレジスタ23に乱数が書き込まれるような回路構成で実現できる。

【 0 0 3 9 】

以上のような対策を施した暗号演算回路とすることにより、Lレジスタ23の値の遷移を攻撃のポイントとできなくなった。

【 0 0 4 0 】

次に、図10のストレート型回路構成の場合についてのDPA法による攻撃のポイントとなるタイミングとその原因について説明する。

【 0 0 4 1 】

ストレート型回路構成においてもクロス型回路構成のときと同様に、演算結果を出力するときのクロック $Clock_{fin}$ の立ち上がりのタイミングにおいて、Lレジスタ23の値を L_{fin} 、Rレジスタ22の値を R_{fin} とする。 $Clock_{fin}$ の次のクロック $Clock_{fin+1}$ の立ち上がりのタイミングにおいて、Lレジスタ23、Rレジスタ22の値はそれぞれ L_{fin+1} 、 R_{fin+1} とすると、次式で表すことができる。

$$L_{fin+1} = L_{fin} \wedge F(key_{fin}, R_{fin})$$

$$R_{fin+1} = R_{fin}$$

この $Clock_{fin+1}$ のタイミングにおいて、Lレジスタ23の値の遷移に着目すると、Lレジスタ23の値の遷移は $L_{fin} \wedge L_{fin+1} = F(key_{fin}, R_{fin})$ と表すことができる。Lレジスタ23の値の遷移 $F(key_{fin}, R_{fin})$ は、鍵情報に関する情報を含んでいるうえに、 R_{fin} は外部から観測可能な値である。従って、Lレジスタ23の値の遷移に対し、DPA法を用いて攻撃が行われると、暗号演算回路の鍵情報 key_{fin} が特定されてしまう。

【 0 0 4 2 】

そのため、本実施の形態においては、Lレジスタ23の値が、 $Clock_{fin+1}$ において暗号演算回路が持つ鍵情報と相関を持つことのないような回路構成とする。この回路構成の具体例としては、クロス型回路構成で説明した方法と同様の図12にあるようにLレジスタ23へのデータパス28上にAND、OR、XORなどの論理演算器を配置し、データパス28の値と乱数の演算結果をLレジスタ23に書き込むという方法や、図11にあるように、 $Clock_{fin+1}$ のタイミングにて、データパス28の値と乱数とをセクタで切り替えるという方法で実現可能である。

【 0 0 4 3 】

また、関数演算部24に $Clock_{fin+1}$ 以降、常に key_{fin} と R_{fin} が入力されるような回路構成の場合には、上記のクロス型回路構成の時と同様に関数演算部24の出力値の遷移によって鍵情報が特定される可能性がある。そのため、前記クロス型回路構成の関数演算部24のDPA法の対策と同様、図4～図9の何れかを用いることによって、対策を実現できる。

【 0 0 4 4 】

また、クロス型回路構成、ストレート回路構成の何れの場合においても、 $Clock_{fin+1}$ においてRレジスタ22、Lレジスタ23へのデータの書き込みが行なわれず、且つ、 $Clock_{fin+1}$ 以降、常に鍵情報 key_{fin} が関数演算部24に入力されるような回路構成が取られている場合には、関数演算部24の出力値 $F(key_{fin}, R_{fin})$ となるので、上記のクロス型回路構成の時と同様、関数演算部24の出力値の遷移がDPA法による攻撃の対象となってしまう。従って、上記のクロス型回路構成の関数演算部24のDPA法の対策と同様、図4～図9の何れかを用いることによって、対策を実現できる。

【 0 0 4 5 】

以上説明してきたように、本実施の形態では、暗号演算回路の演算終了後以降において、鍵情報と消費電力の相関関係を失わせるような回路構成としたから、暗号演算回路内部にある鍵情報の漏洩を防ぐことができ、DPA法による攻撃に対する耐性を持った暗号演算回路を提供できるようになった。

【 0 0 4 6 】

【発明の効果】

以上説明したように、本発明によれば、暗号演算回路内部にある鍵情報の漏洩を防ぐことができ、DPA法による攻撃に対する耐性を持った暗号演算回路を提供できるようになった。

【図面の簡単な説明】

【図 1】 Feistel型アルゴリズムを示すフローチャート。

【図 2】 Feistel型アルゴリズムを採用した暗号演算回路のデータパス部に関する回路構成を示す図。

【図 3】 クロス型回路構成のデータパス部を示す図。

【図 4】 本実施の形態における、関数演算部 2 4 の D P A 法対策を施した回路構成の第 1 例を示す図。

【図 5】 本実施の形態における、関数演算部 2 4 の D P A 法対策を施した回路構成の第 2 例を示す図。

【図 6】 本実施の形態における、関数演算部 2 4 の D P A 法対策を施した回路構成の第 3 例を示す図。

【図 7】 本実施の形態における、関数演算部 2 4 の D P A 法対策を施した回路構成の第 4 例を示す図。

【図 8】 本実施の形態における、関数演算部 2 4 の D P A 法対策を施した回路構成の第 5 例を示す図。

【図 9】 本実施の形態における、関数演算部 2 4 の D P A 法対策を施した回路構成の第 6 例を示す図。

【図 1 0】 ストレート型回路構成のデータパス部を示す図。

【図 1 1】 本実施の形態における、Lレジスタ 2 3 の D P A 法対策を施した回路構成の第 1 例を示す図。

【図 1 2】 本実施の形態における、Lレジスタ 2 3 の D P A 法対策を施した回路構成の第 2 例を示す図。

【符号の説明】

2 1 … I P 部

2 2 … R レジスタ

2 3 … L レジスタ

2 4 … 関数演算部

2 5 … IP^{-1} 部

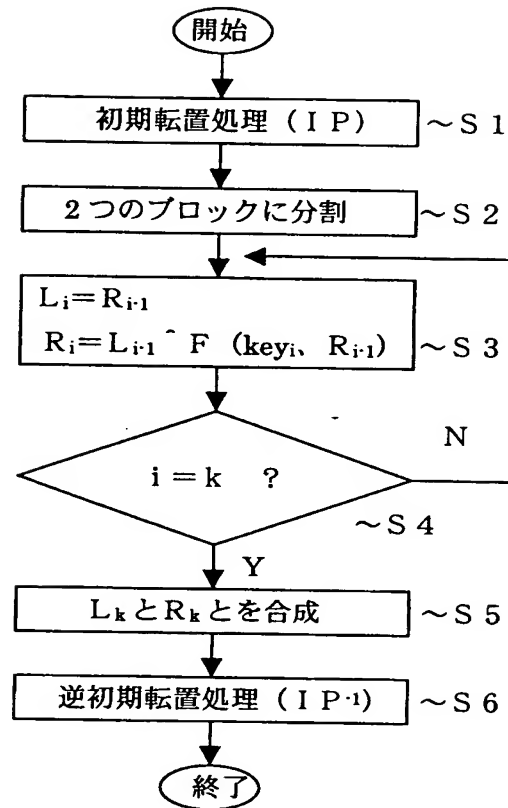
2 6、2 7 … セレクタ

2 8、2 9、3 0 … データパス

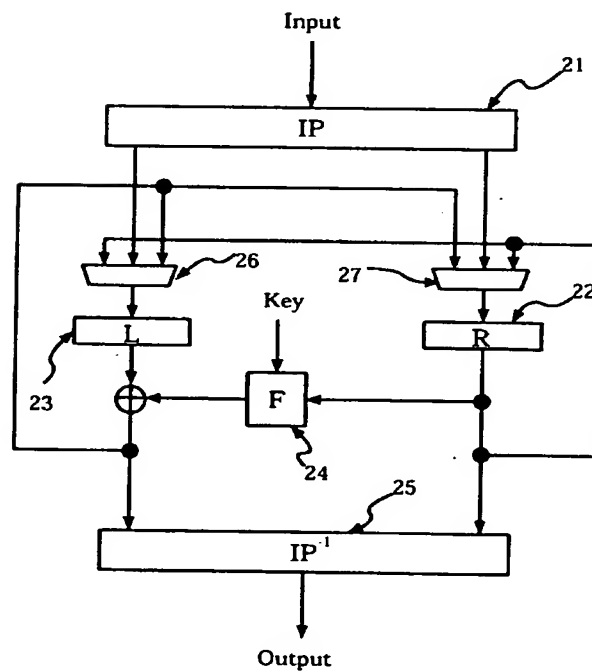
【書類名】

図面

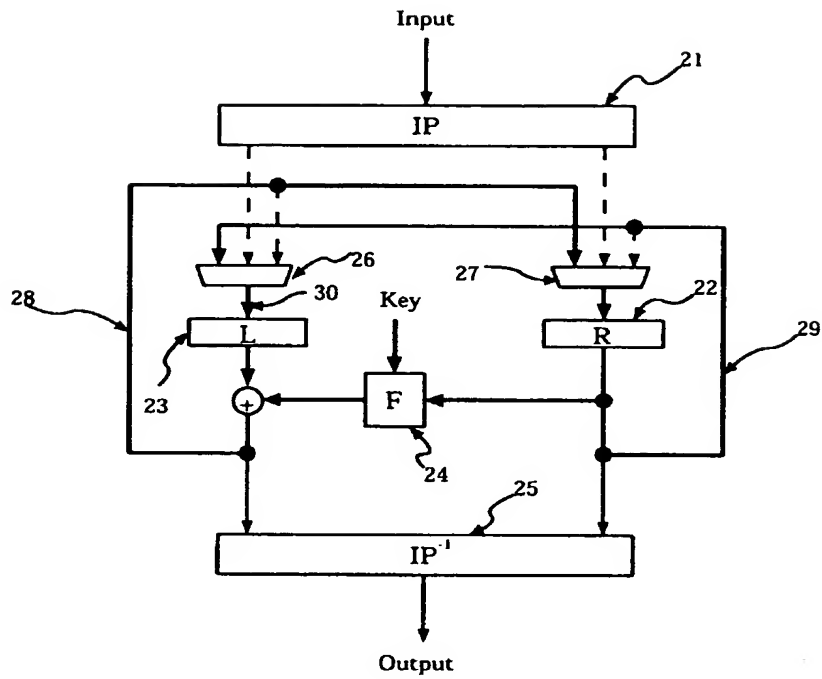
【図 1】



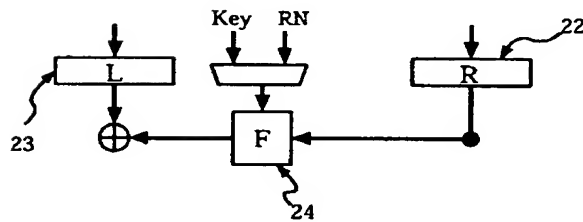
【図 2】



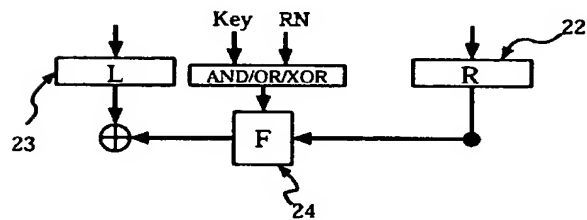
【图 3】



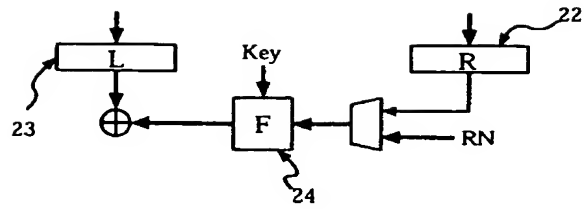
【図 4】



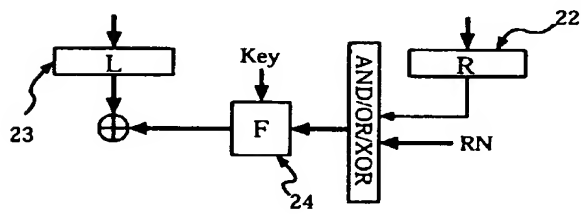
【図 5】



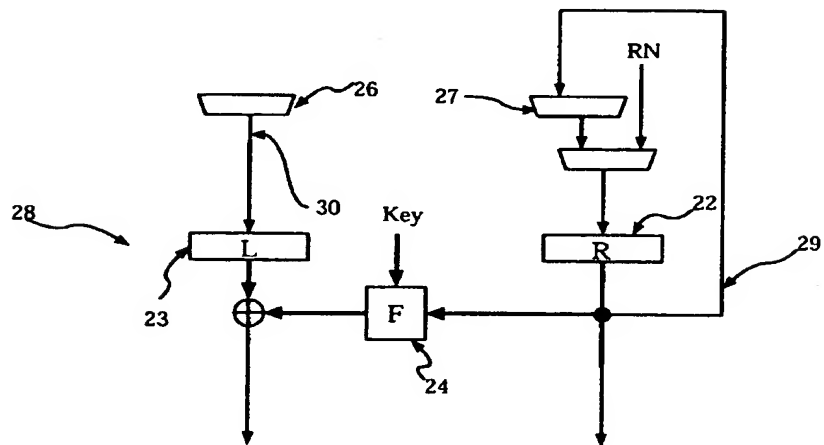
【図 6】



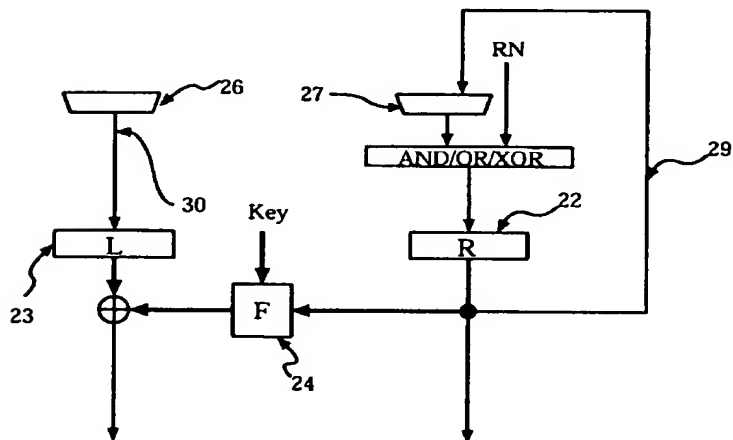
【図 7】



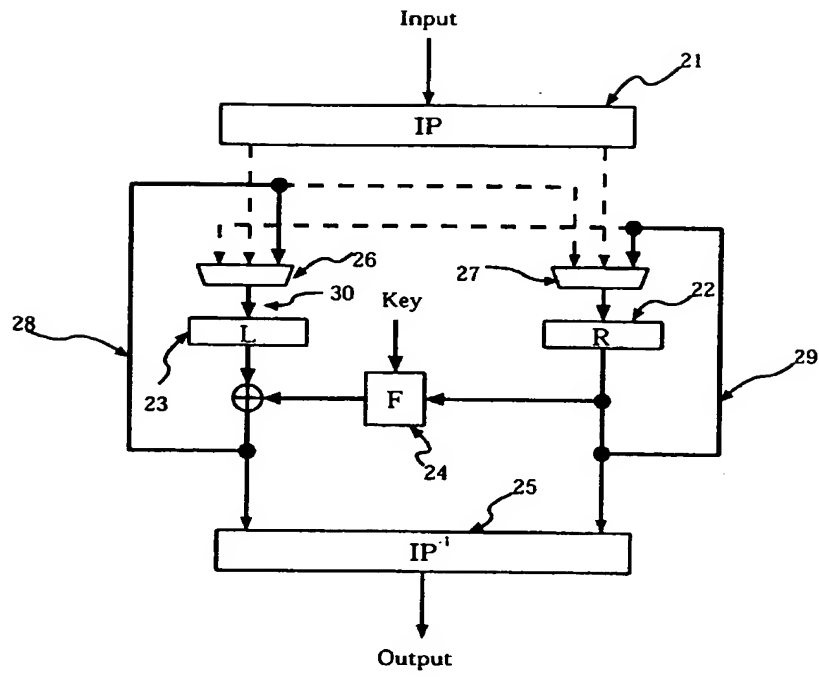
【図 8】



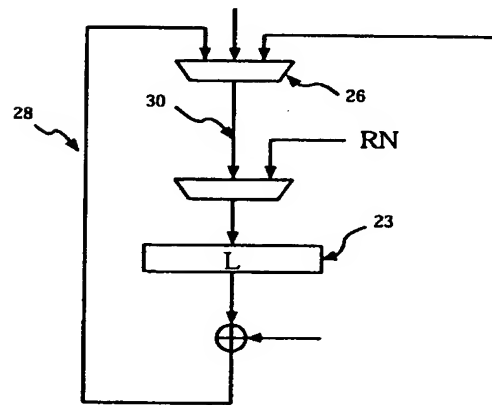
【図 9】



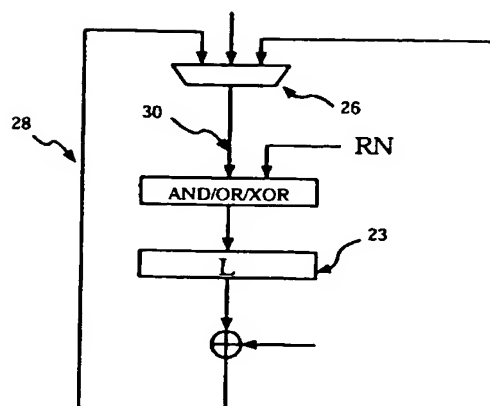
【図 1 0】



【図 1 1】



【図 1 2】



【書類名】 要約書

【要約】

【課題】 Feistel 型暗号アルゴリズムを実装した暗号演算回路の演算終了の後に行われる DPA 法の対策を実現した暗号演算回路の提供。

【解決手段】 Feistel 型暗号アルゴリズムを実装した暗号演算回路は、非線形関数 F を演算する関数演算部 24 と、暗号化データの出力の後に、関数演算部 24 へ暗号演算結果とは無関係の乱数データを供給する手段とを備える。

【選択図】 図 4

認 定 ・ 付 加 情 報

特許出願の番号	特願 2 0 0 2 - 2 6 4 9 7 7
受付番号	5 0 2 0 1 3 5 7 0 8 6
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 4 年 9 月 1 2 日

< 認定情報・付加情報 >

【提出日】	平成14年 9月11日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 3 0 7 8]

1. 変更年月日	2 0 0 1 年 7 月 2 日
[変更理由]	住所変更
住 所	東京都港区芝浦一丁目 1 番 1 号
氏 名	株式会社東芝